

Reconocimiento de Patrones

Version 2022-2

Gabor

[Capítulo 2]

Dr. José Ramón Iglesias

DSP-ASIC BUILDER GROUP

Director Semillero TRIAC

Ingeniería Electronica

Universidad Popular del Cesar

Texture Features for Browsing and Retrieval of Image Data

B.S. Manjunath and W.Y. Ma

Abstract—Image content based retrieval is emerging as an important research area with application to digital libraries and multimedia databases. The focus of this paper is on the image processing aspects and in particular using texture information for browsing and retrieval of large image data. We propose the use of Gabor wavelet features for texture analysis and provide a comprehensive experimental evaluation. Comparisons with other multiresolution texture features using the Brodatz texture database indicate that the Gabor features provide the best pattern retrieval accuracy. An application to browsing large air photos is illustrated.

Index Terms—Digital libraries, image database, content-based image retrieval, texture analysis, Gabor wavelets.

Funciones de Gabor

The Gabor functions are a complete (but a nonorthogonal) basis set given by:

$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right) \exp(2\pi j u_0 x)$$

where σ_x and σ_y denote the Gaussian envelope along the x and y -axes, and u_0 defines the radial frequency of the Gabor function.

Funciones de Gabor

self-similar filter bank can be obtained by appropriate dilation and rotation of $f(x, y)$ through the generating function

$$f_{pq}(x, y) = \alpha^{-p} f(x', y')$$

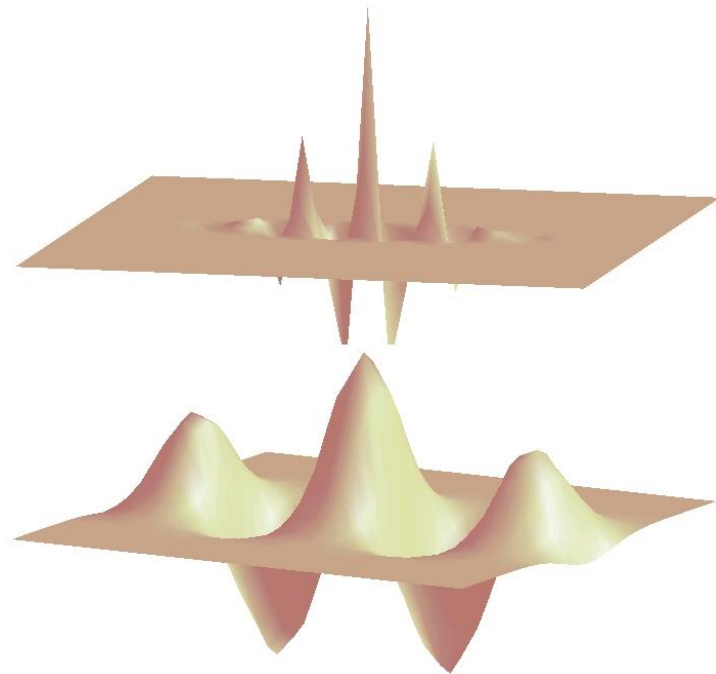
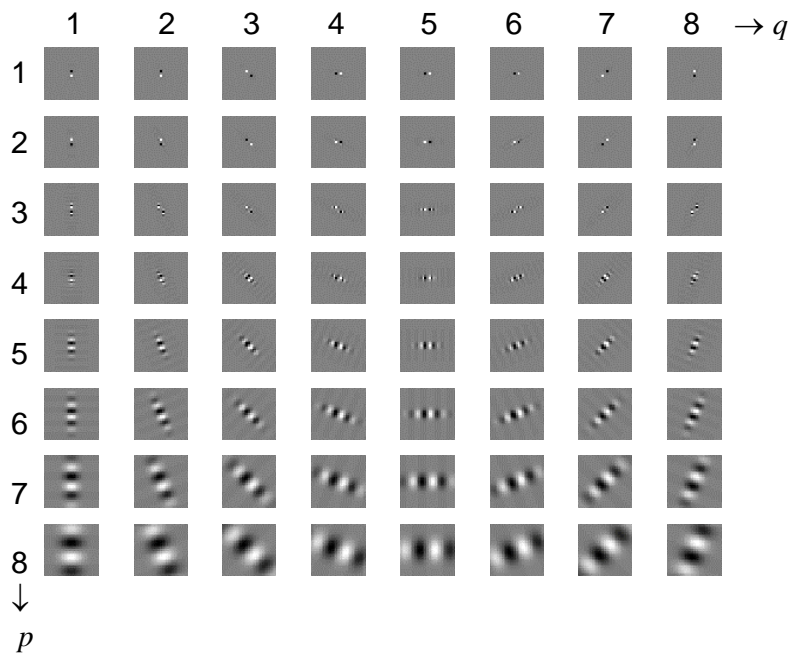
where

$$\begin{aligned} x' &= \alpha^{-p}(x \cos \theta_q + y \sin \theta_q) \\ &= \alpha^{-p}(-x \sin \theta_q + y \cos \theta_q), \\ \alpha &> 1; \quad p = 1, 2, \dots, S; \quad q = 1, 2, \dots, L. \end{aligned}$$

The integer subscripts p and q represent the index for scale (dilation) and orientation (rotation), respectively. S is the total number of scales and L is the total number of orientations in the self-similar Gabor filter bank. For each orientation q , the angle θ_q is given by

$$\theta_q = \frac{\pi(q-1)}{L}, \quad q = 1, 2, \dots, L.$$

Funciones de Gabor



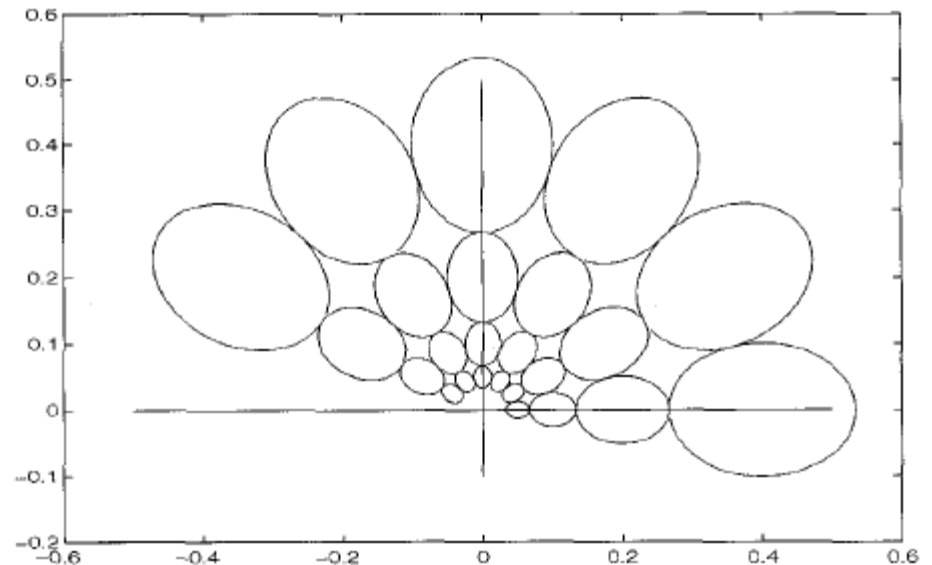
Funciones de Gabor

$$\alpha = \left(\frac{f_h}{f_l} \right)^{-(1/(S-1))}$$

$$\sigma_x = \frac{\sqrt{2 \ln 2}(\alpha + 1)}{2\pi f_h(\alpha - 1)}$$

$$\sigma_y = \left[2 \ln 2 - \left(\frac{2 \ln 2}{2\pi \sigma_x f_h} \right)^2 \right]^{1/2} \cdot \left[2\pi \tan\left(\frac{\pi}{2L}\right) \left(f_h - 2 \ln\left(\frac{1}{4\pi^2 \sigma_x^2 f_h}\right) \right) \right]^{-1}$$

Se asegura así que la respuesta en frecuencia de los filtros Gabor escogidos no se traslapen.



Funciones de Gabor

$$I_{pq}(x, y) = \left\{ [f_{pq}(x, y)_e * I(x, y)]^2 + [f_{pq}(x, y)_o * I(x, y)]^2 \right\}^{1/2} \quad (6)$$

where “*” denotes 2-D convolution operation, and $f_{pq}(x, y)_e$ and $f_{pq}(x, y)_o$ represent the even and odd parts of the Gabor filter separated from (3).

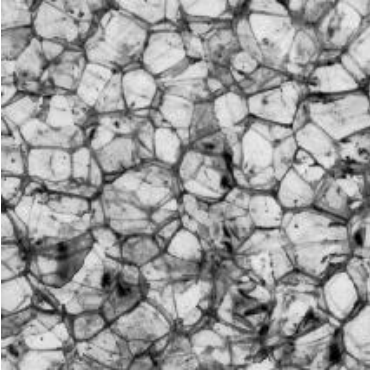
for $p=1 \dots s$ for $q=1 \dots r$ (‘s’ scales and ‘r’ orientations)

$$g(p, q) = I_{pq}$$

$$J = (g_{\max} - g_{\min}) / g_{\min}$$

Typically, $r = s = 8$

Input



I

Kernels Gabor

G_{11}



G_{12}



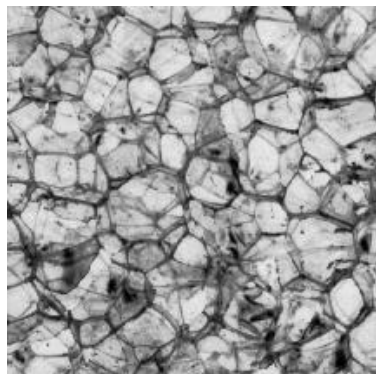
:



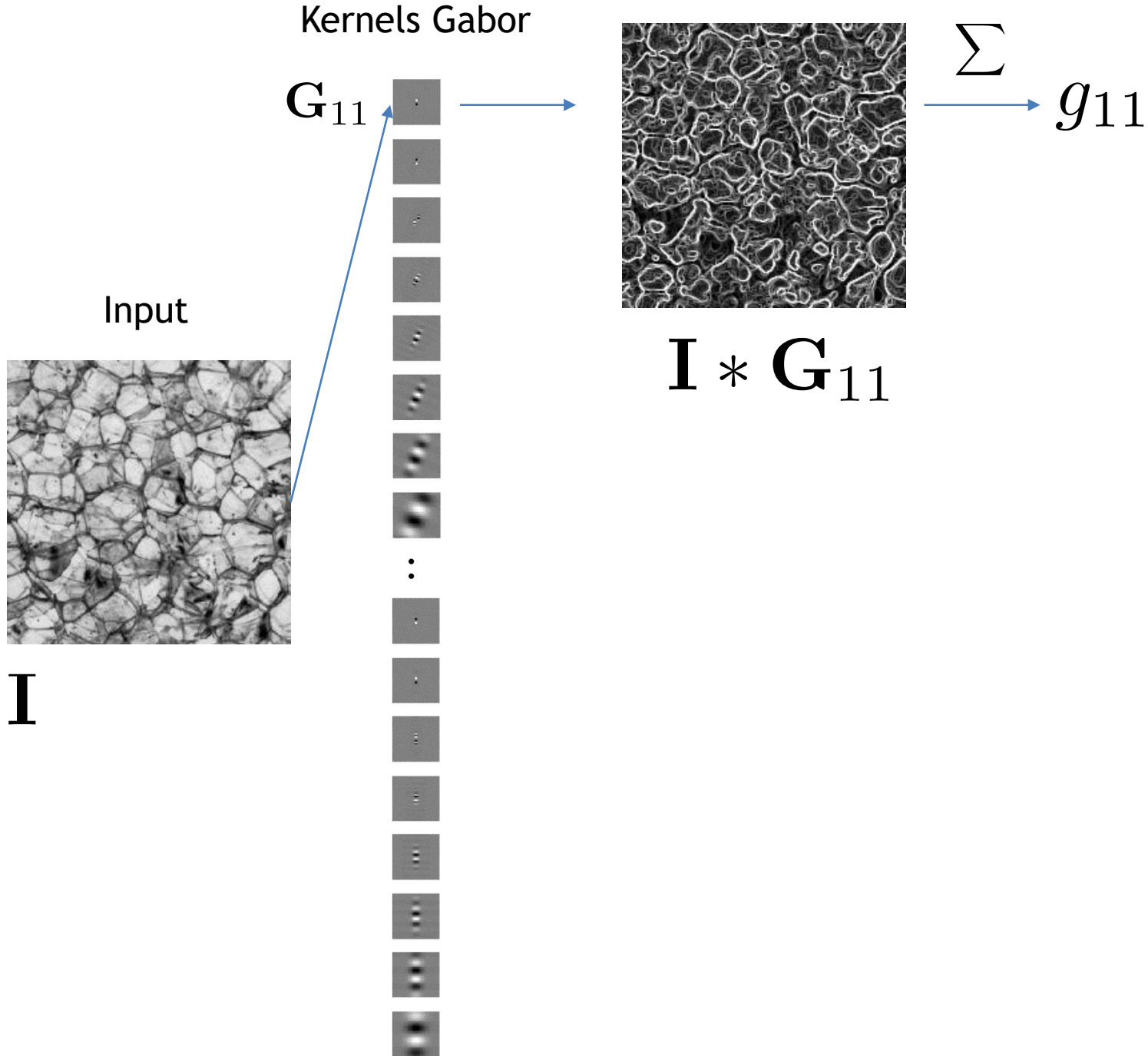
G_{88}

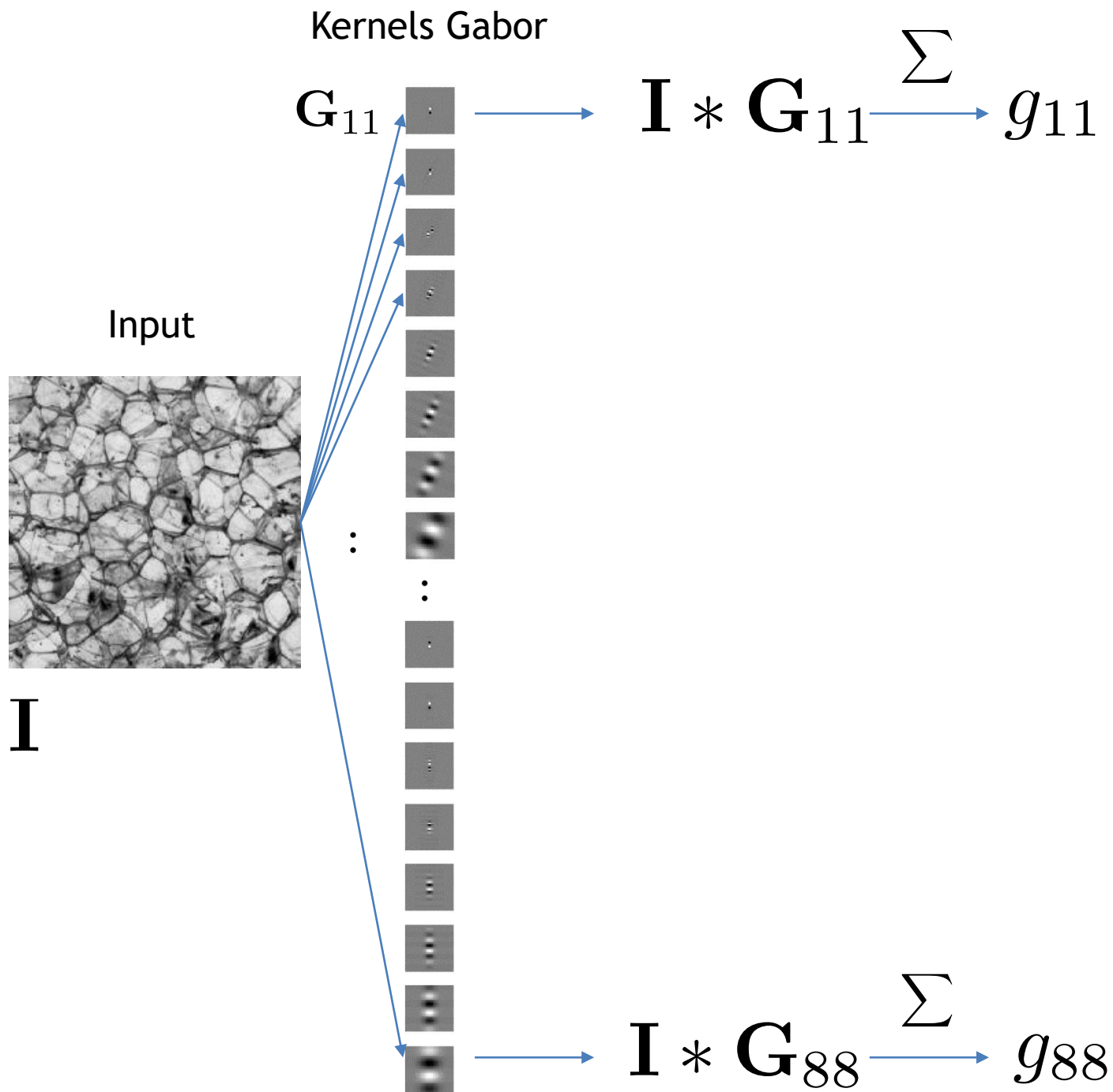


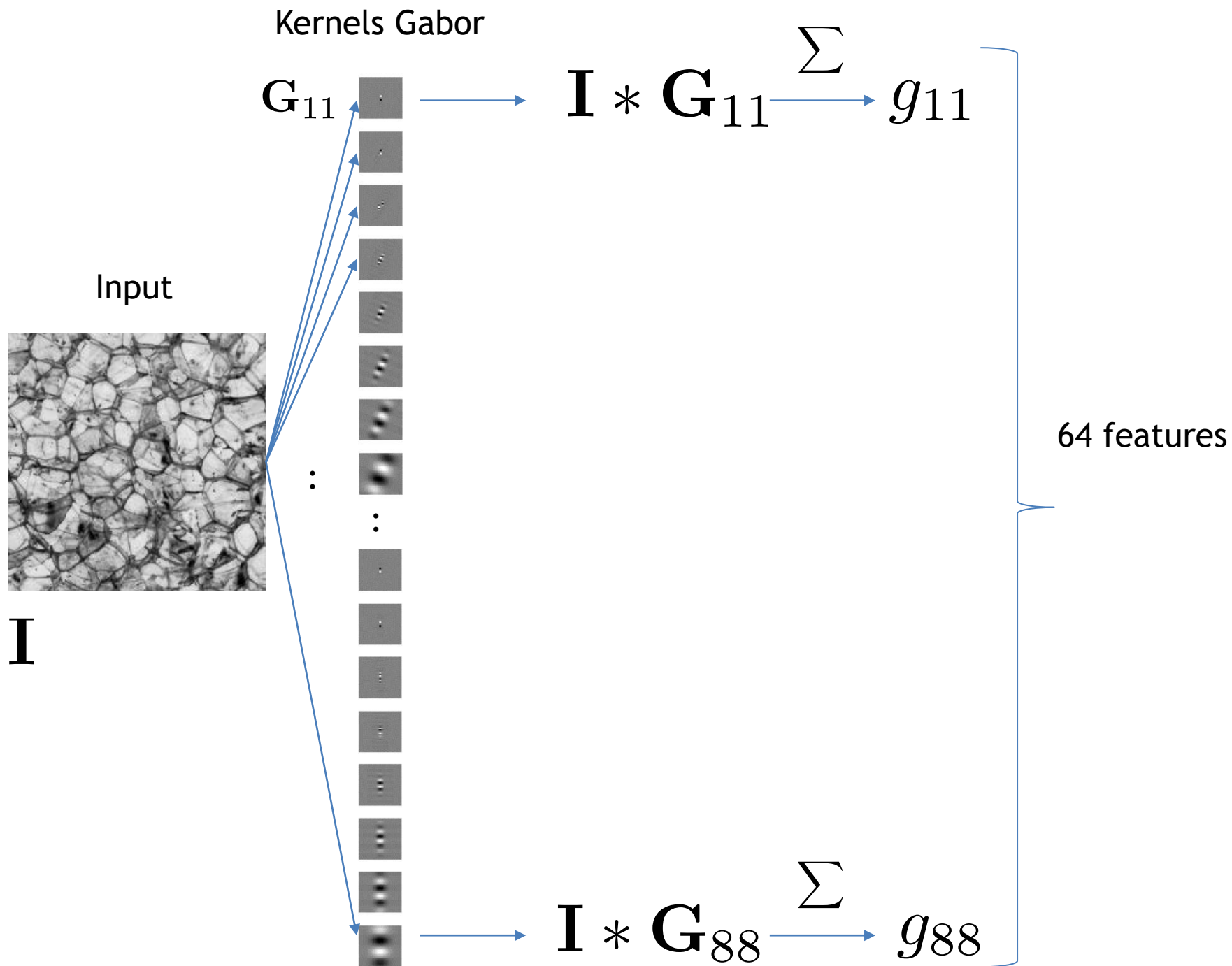
Input

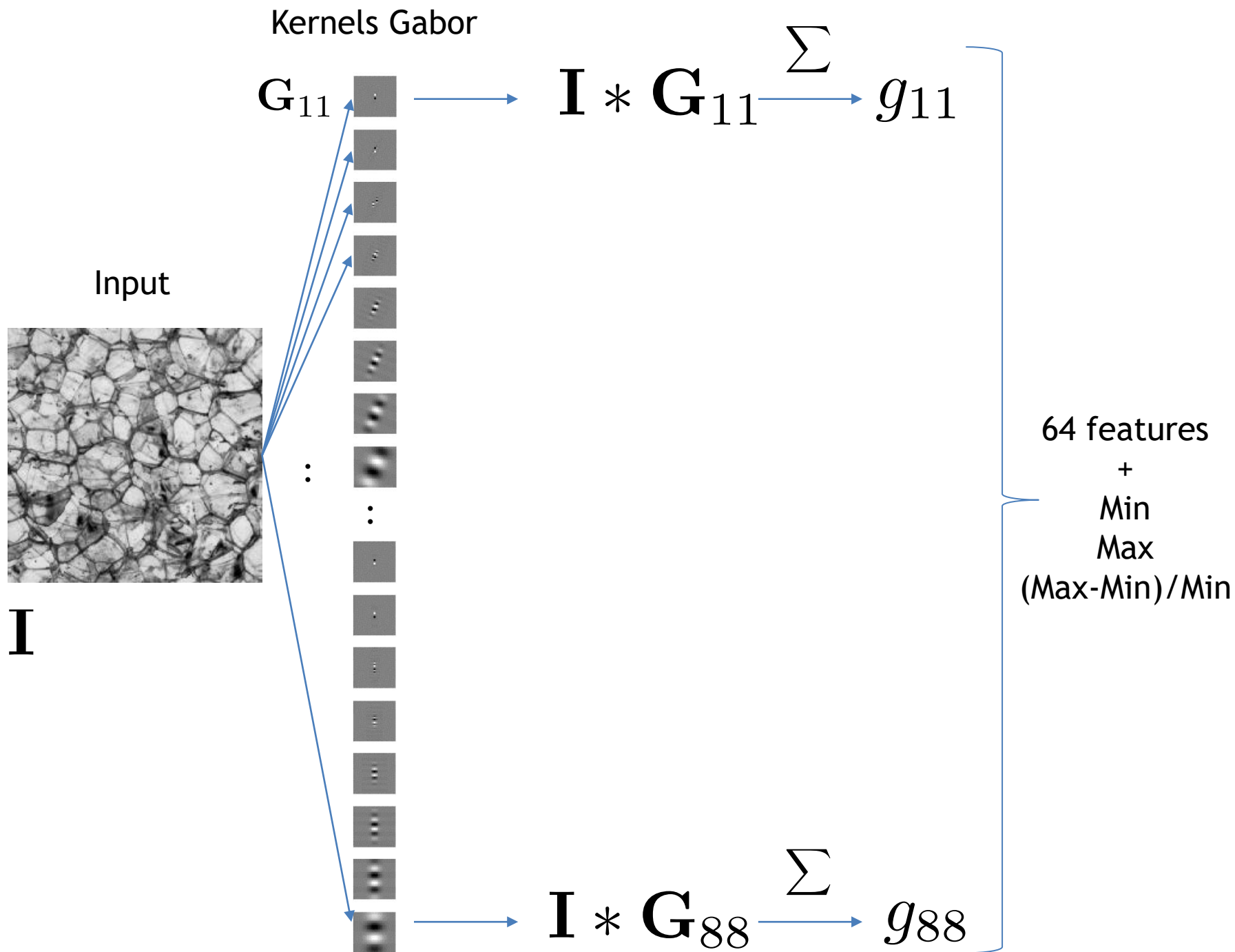


I



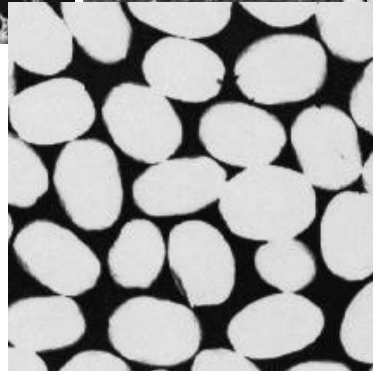
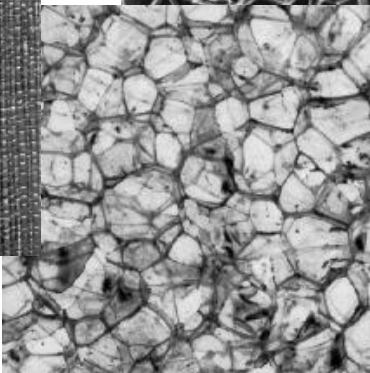
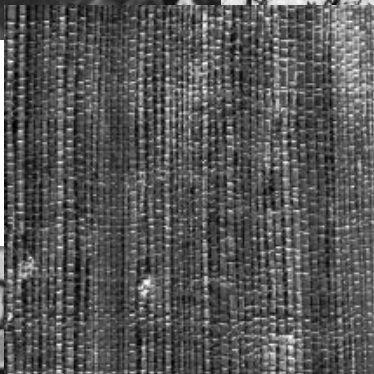
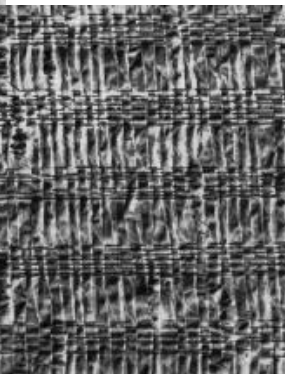
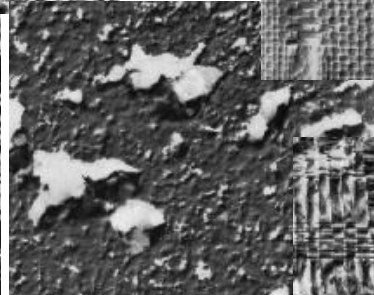
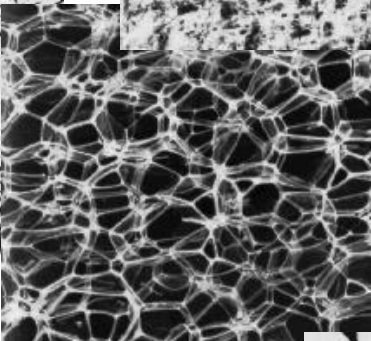
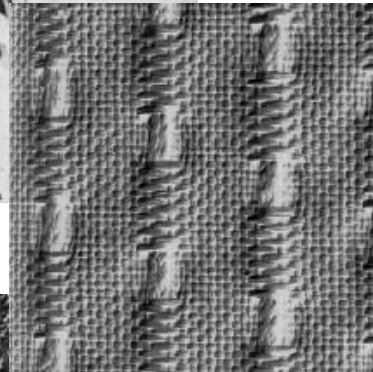
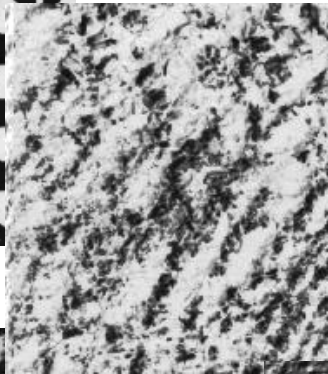
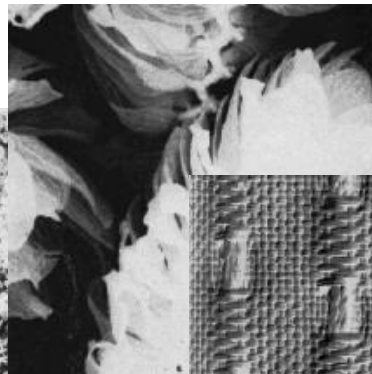
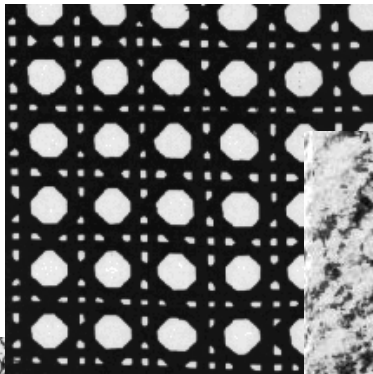
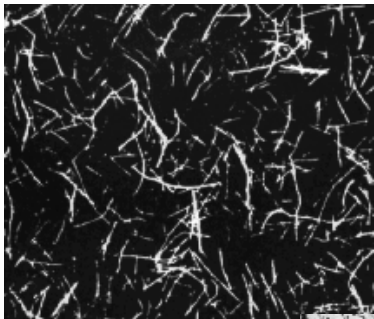


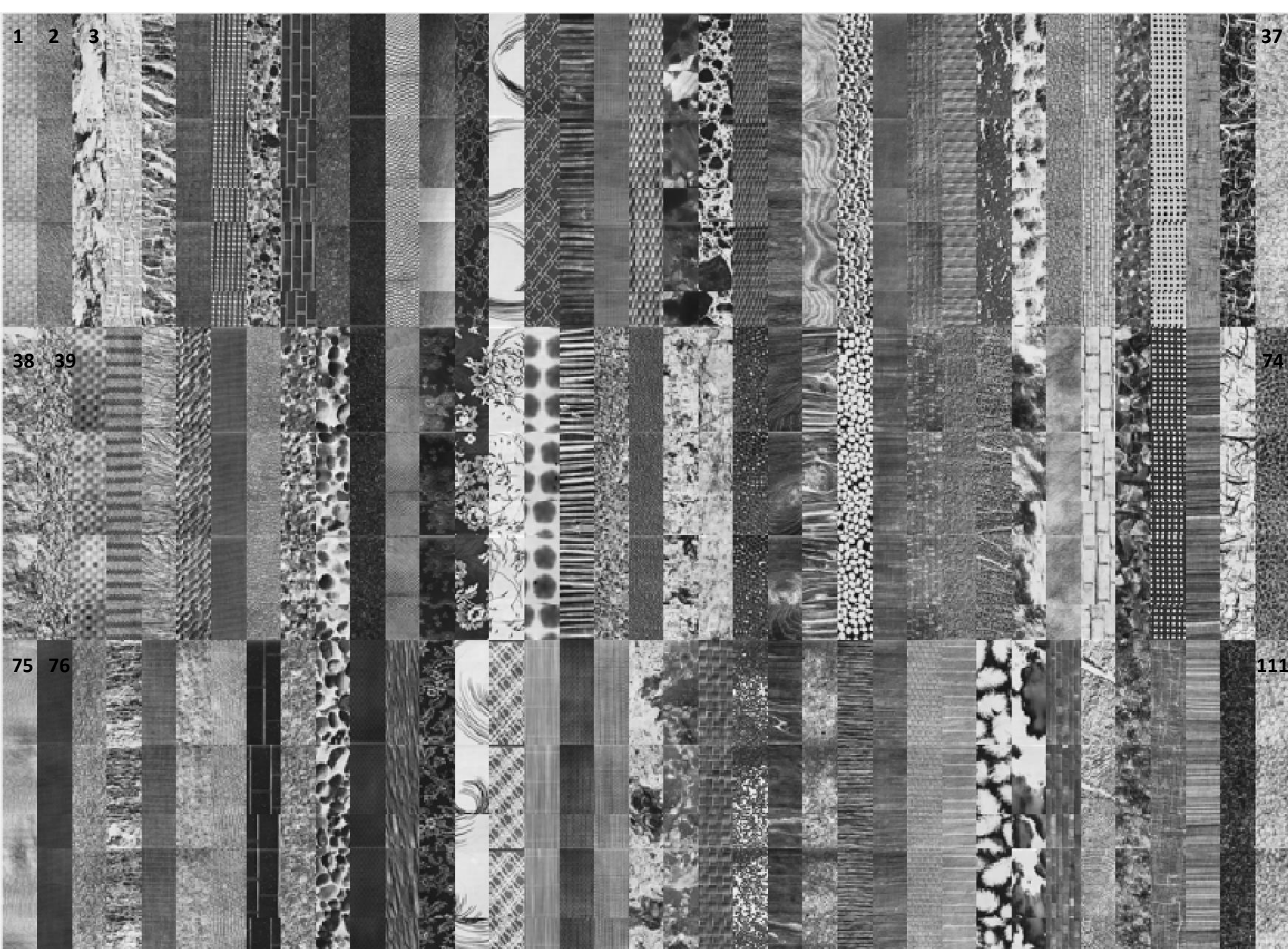




Ejemplo

Base de Datos: Brodatz
111 texturas
9 imágenes por textura





Cómo diseñar un clasificador que reconozca 111 Texturas?

Solución (1 / 2)

1. Extracción de Características

BD: 111 clases,
9 imágenes por clase

→ Gabor →

Matriz X: 999 x 67 elementos
Vector y: 999 elementos

2. Separación de Training/Testing

Training: características de las primeras 8 imágenes por clase

Xtrain: matriz de 888 x 67 ytrain: vector de 888 elementos

Testing: características de la última imagen de la clase

Xtest: matriz de 111 x 67 ytest: vector de 111 elementos

Solución (2/2)

3. Diseño del Clasificador



4. Predicción en el set de testing



5. Evaluación de Desempeño



Resultados (KNN con $k=1$)

